Explaining predictions made by neural networks

Monday, 13th of May, 2019

Context

1. Problem Statement
2. Approaches
3. Related Work
4. Some of the modern approaches

Problem Statement

To explain the outcome of a deep neural network model.

Approaches

1. **No modifications to the existing architectures :**

On a previously trained model, estimating the concepts/explanations by propagating in the reverse direction of the model.

2. **Modifications to the existing architectures :**

To develop models where interpretability is built-in the architecture itself. These models can explain the concepts during learning.

Related Work

1. Approach 1 - Based on Input Influence

It is based on mapping models’ prediction outputs back to relevant regions in an input image i.e. estimates the relevance of input features in the predicted output.

- Sensitivity Analysis

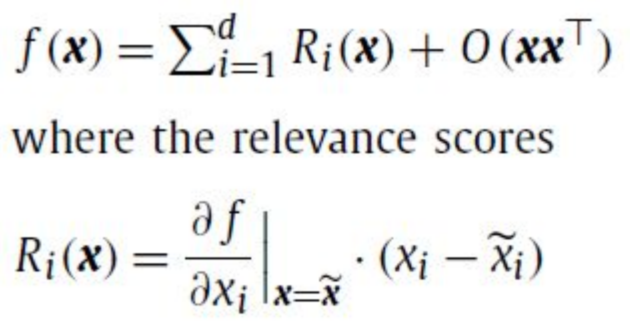
It estimates the relevance score of input image features wrt to the local gradient of the output function and not the function itself. And thus the heatmap generated using this method on the input image do not focus on the actual class relevant features.

$$\sum_{i=1}^{d} R_i(\boldsymbol{x}) = \|\nabla f(\boldsymbol{x})\|^2.$$

K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, CoRR, arXiv:1312.6034, 2013.

- Taylor Decomposition

It explains the model’s decision by decomposing the function value f(x) as a sum of relevance scores. The relevance scores are obtained by identification of the terms of a first-order Taylor expansion of the function at some root point x~ for which f(x~) = 0.



$$f(\boldsymbol{x}) = \sum_{i=1}^{d} R_i(\boldsymbol{x}) + O(\boldsymbol{x}\boldsymbol{x}^\top)$$

where the relevance scores

$$R_i(\boldsymbol{x}) = \frac{\partial f}{\partial x_i}\bigg|_{\boldsymbol{x}=\widetilde{\boldsymbol{x}}} \cdot (x_i - \widetilde{x}_i)$$

For higher order terms to be zero, x~ must be chosen closer to x. So, this decompositions provides partial information about the f(x) except in some special cases when the function is linear and satisfies the required property.

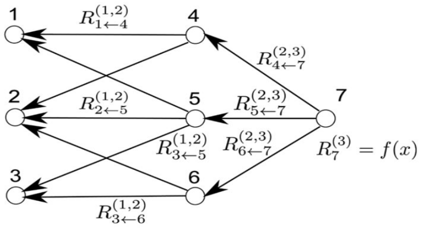
- Layer - Wise Relevance Propagation (LRP) :

It estimates relevance scores of each input feature wrt the output function value which is also considered as the relevance score of output neuron.

It is done by backpropagating relevance scores layerwise in such a way that sum of the relevance score at each layer is conserved.

$$f(x) = \cdots = \sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l} R_d^{(l)} = \cdots = \sum_{d} R_d^{(1)}$$

Let j and k be indices for neurons of two successive layers. Let Rk be the relevance of neuron k for the prediction f(x). We define Rj←k as the share of Rk that is redistributed to neuron j in the lower layer.


1
$R_{1\leftarrow 4}^{(1,2)}$
4
$R_{4\leftarrow 7}^{(2,3)}$
2
$R_{2\leftarrow 5}^{(1,2)}$
5
$R_{5\leftarrow 7}^{(2,3)}$
7
$R_{3\leftarrow 5}^{(1,2)}$
$R_{6\leftarrow 7}^{(2,3)}$
$R_7^{(3)} = f(x)$
3
6
$R_{3\leftarrow 6}^{(1,2)}$

The conservation property for this neuron imposes

$$\sum_j R_{j\leftarrow k} = R_k. \qquad \text{------ (1)}$$

Likewise, neurons in the lower layer aggregate all relevance coming from the neurons from the higher layer:

$$R_j = \sum_k R_{j\leftarrow k} \qquad \text{----- (2)}$$

An algorithm would start with relevances R(l+1) of layer l+1 which have been computed already. Then the messages $R_{j\leftarrow k}$ would be computed for all elements k from layer l+1 and elements i from the preceding layer l, in a manner such that Eq (1) holds. Then definition (2) would be used to define the relevances R (l) for all elements of layer l.

One propagation rule that is locally conservative and that was shown to work well in practice is the αβ-rule given by:

$$R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)} \cdot \left(\alpha \cdot \frac{z_{ij}^+}{z_j^+} + \beta \cdot \frac{z_{ij}^-}{z_j^-}\right)$$

Let $z_j^+ = \sum_i z_{ij}^+ + b_j^+$ and $z_j^- = \sum_i z_{ij}^- + b_j^-$

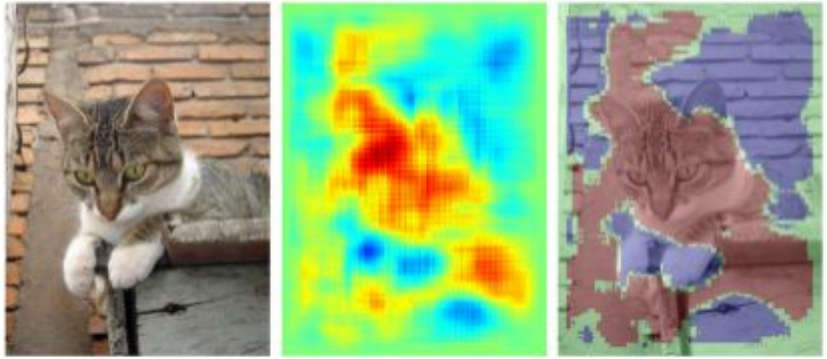


Fig : Pixel-wise decomposition for features over a histogram intersection kernel using the layer-wise relevance propagation for all subsequent layers and rank-mapping for mapping local features. Each triplet of images shows—from left to right—the original image, the pixel-wise predictions superimposed with prominent edges from the input image and the original image superimposed with binarized pixel-wise predictions. The decompositions were computed on the whole image. Positive responses seem to exist for certain fur texture patterns, see also the false responses on the wood and the plaster in the second example which both have similar texture and color to a cat’s fur.

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Mller, and W. Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLOS ONE, 10(7):0130140, 2015.

2. **Approach 1 - Based on Internal Influence**

The above approaches relate instance-specific features to instance-specific predictions, explanations that they produce do not generalize beyond a single test point.

K. Leino, L. Li, S. Sen, A. Datta, and M. Fredrikson. Influence-Directed Explanations for Deep Convolutional Networks. In CoRR abs/1802.03788, 2018.

This method promises to

(1) identify influential concepts that generalize across instances,

(2) can be used to extract the “essence” of what the network learned about a class, and

(3) isolate individual features the network uses to make decisions and distinguish related classes.

It defines *Distributional influence* which is parameterized by a *slice s* of the network, a *quantity of interest f* and a *distribution of interest P* . Given these elements, we measure influence as the partial derivative of f at the slice s averaged over P.

The influence of an element (neuron) j in the internal representation is defined as follows:

$$\chi_j^s(f, P) = \int_{\mathcal{X}} \left.\frac{\partial g}{\partial z_j}\right|_{h(\mathbf{x})} P(\mathbf{x}) d\mathbf{x}$$

The *slice parameter* exposes the internals of a network, allowing us to measure influence with respect to intermediate neurons and identifying high-level concepts that are learned by a network.

distributions of interest are: (1) a single instance - focuses on why a single instance was classified a particular way,

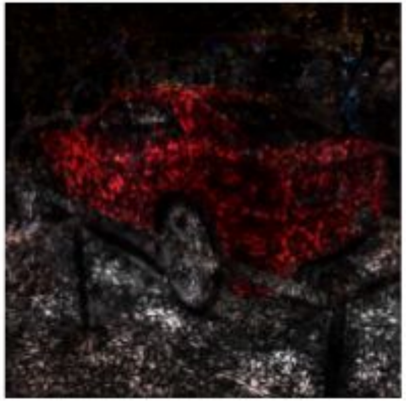
(2) the distribution of ‘cat’ images - the second explains the “essence” of a class, or

(3) the distribution of all images in a dataset - identifies generally-influential neurons over the entire population.

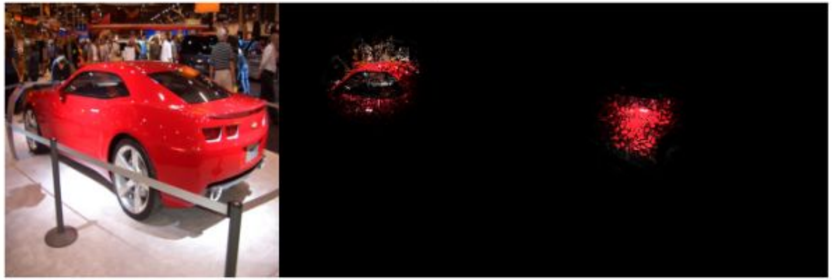
the *quantity of interest* may correspond to

(1) the network’s outcome for the ‘cat’ - addresses the question of why a particular input is classified as ‘cat’

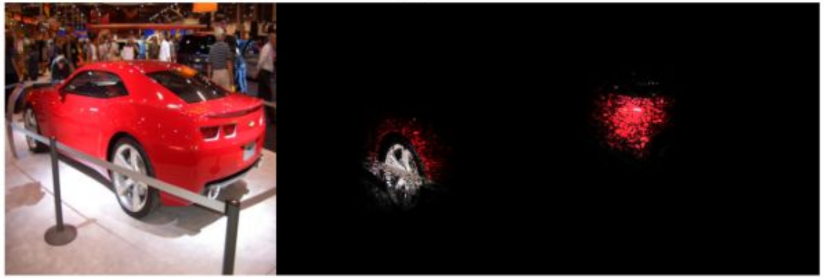
(2) or its comparative outcome towards ‘cat’ versus ‘dog’ (i.e.,the difference in the network scores for cat and dog classes). - addresses how the network distinguishes ‘cat’ instances from ‘dog’ instances



-> explanation provided by input influence



- > two most influential units for the quantity of interest characterizing correct classification (sports car)



-> top two most influential units which is understood to be its most distinctive feature according to this comparative quantity (convertible).

3. **Approach 2 - Self Explaining Models**

David Alvarez-Melis, Tommi S. Jaakkola, Towards Robust Interpretability with Self-Explaining Neural Networks

Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu University of California, Los Angeles, Interpretable Convolutional Neural Networks

Thank You